Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

Software Vulnerability Disclosure and its Impact on Exploitation: An Empirical Study

George A. Mangalaraj
The University of Texas at Arlington, magalaraj@uta.edu

M. K. Raja
University of Texas at Arlington, raja@uta.edu

Software vulnerability disclosure and its impact on exploitation: An empirical study

George A. Mangalaraj
The University of Texas at Arlington
mangalaraj@uta.edu

M. K. Raja
The University of Texas at Arlington
raja@uta.edu

ABSTRACT

In a networked world, computer systems are highly exposed to the attacks of worms / viruses. Many of these attacks stem from the vulnerabilities in the software code. One of the issues that plagues the information security area is the publicly available information about the vulnerabilities in popular software applications. This information has been put to good as well as bad use by people in the technical community. Software vendors and the anti-virus companies develop patches to resolve the software vulnerability. Hackers and virus writers make use of the same information to write malicious code to exploit the vulnerability. This exploratory study analyzes whether the information availability has an impact on the exploitation of the vulnerability. This study also considers some of the characteristics of the vulnerability information and its impact on the exploitation. Two of the factors thus considered, namely, the criticality, and cumulativeness of the vulnerability was found to have a significant impact on the actual exploitation.

Keywords

Software vulnerability, information disclosure, exploitation, virus, worms, malicious code

INTRODUCTION

Last few years saw a string of attacks on computers that relied on virus / worms. Many of these viruses / worms utilized the vulnerabilities of software application. Software vulnerability information is widely available on the Internet through various sources. These vulnerabilities are exploited by malicious code writers to create malware such as viruses and worms. Annually billions of dollars are lost in these attacks (CSI-FBI 2004). This situation is exacerbated by the pervasiveness of computer and the communication networks. Objective of this study is to analyze the impact of announcing vulnerabilities by the software vendors due to the potential misuse by the malicious code writers. Determinants for the exploitation of vulnerability are arrived at based on the information availability and the characteristics of such information. Hypotheses were formulated in this regard and tested using data from publicly available databases.

Following is the outline of this paper; next section reviews the literature on software vulnerability and issues related to it. The section that follows proposes various hypotheses based on the characteristics of vulnerability. The subsequent section outlines the data collection procedure and presents the results. The final section concludes this study with the limitations and areas for future research.

LITERATURE REVIEW

Software Vulnerability

Information security issues related to the vulnerabilities of software products are increasingly gaining attention. Krsul (1998) define software vulnerability as:

> "an instance of an error in the specification, development, or configuration of software such that its execution can violate the security policy".

This definition implies that flaws in software applications cause the vulnerability to exist. Fithen, Hernan et al.(2003) define vulnerability as an unplanned system feature that an intruder may exploit if there were certain preconditions to achieve particular impacts on that system in violation of its security policy. This definition identifies software flaws and the existence of preconditions in the software application as a source for software vulnerability. Fithen, Hernan et al.(2003) also introduce

the role of an intruder in exploiting the vulnerabilities. Both the definitions imply that software vulnerabilities have information security implications. These software vulnerabilities are needed to be addressed before an attacker exploits it.

In organizations, software products are either developed by in-house Information Systems department or developed by outside software vendors. Increasingly, organizations are relying on the software developed by other companies outside their organizations and it can come from commercial off-the-shelf (COTS) software vendors or the providers of open-source software. Vulnerability information about COTS software products are hard to obtain and this has led to the formation of the Common Vulnerabilities and Exposure (CVE) initiative (Martin 2002) which tries to make the process of finding and fixing of software vulnerabilities easier. This CVE initiative maintains a comprehensive database with all reported vulnerabilities in different software applications.

Since the vulnerability information is publicly available, the possibility of this information being misused is high. Hackers / attackers can utilize this public information to write viruses, worms, and Trojan horses.

Viruses and Worms

Computer viruses and worms have come a long way from its evolution. Initial computer viruses were self-replicating computer programs that propagated by attaching itself to executable files or other files (Nachenberg 1997). Anti-virus companies tried to keep pace with various strains and types of computer viruses. Virus authors became more sophisticated by developing polymorphic viruses that mutated each time it infected a new program. These mutated strains looked different from the original virus and yet caused same kind of damage. Anti-virus programs used newer technologies such as Generic Decryption to detect and remove such polymorphic viruses (Nachenberg 1997).

Recent years saw the emergence of newer breeds of viruses such as macro viruses that utilized the security flaws in the software applications. These viruses / worms differ from the viruses of earlier days which were stand alone programs and were not dependent on any specific software vulnerability to be present. Whitman (2003) in his study on threats to information security, surveyed IT executives and found that, technical software failures and errors to be an important threat to information security.

Vulnerabilities and Viruses

There are arguments for and against disclosing vulnerability information to the general public. Moreover, how much information can be shared with the public is also a contentious issue (Applewhite 2004). There are many arguments in favor of the disclosure of vulnerability information. Firstly, it educates the customers about the flaws in their system. Secondly, it helps to alert the system administrators to install the remedial patches. Thirdly, it assists the companies that are involved in the security to come-up with adequate responses for any anticipated incidents (Goth 2004).

Arguments that are against disclosing vulnerability information primarily revolves around the contention that the vulnerability information may be misused. Hackers can exploit the time lag between the time vulnerability information is made available and the patches developed by the vendors and the application of it by the customers(Furnell 2004). This problem can be illustrated with the incidents that happened in the January 2003, when Slammer worm exploded on the Internet. Microsoft SQL Server 2000, popular database software had buffer overflow vulnerability and Microsoft was notified by an outside organization about the vulnerability in 17th July 2002 and in the subsequent week Microsoft issued a security bulletin with a patch to fix the vulnerability. In January 2003, Slammer worm exploited this buffer overflow vulnerability and caused havoc. This worm took down 90% of all un-patched computers running SQL server software (Panko 2003). In this instance, though the software vendor made the patches available widely, the time lag present in the installation of it by the end users was exploited by Slammer worm writers. Another argument against the disclosure of information about the patches claim that the patches issued by the software vendors can itself be used by the hackers to reverse engineer viruses / worms (Furnell 2004).

RESEARCH HYPOTHESES

Vulnerability Information Disclosure

The controversies in disclosing the presence of software vulnerability raises an important question on the real effect of such disclosures. It would be germane to empirically test the effect of vulnerability disclosure on the actual exploitation. This paper defines *exploitation* as the presence of a known computer virus that targets particular software vulnerability. In the past, there have been studies that compared software vulnerabilities between closed-source and open-source software products (Reinke and Saiedian 2003). Arora, Krishnan et al.(2004) compared patch availability and software vulnerability on the actual exploitation. Venter and Eloff (2004) used historical vulnerability information to forecast vulnerabilities in the

future. Fithen, Hernan et al.(2003) carried out formal modeling of vulnerabilities that tried to map meaningful relationships between multiple vulnerabilities. There has not been any study that specifically explored the relationship between the vulnerability disclosure and the actual exploitation of it. Hence the first hypothesis is formulated to test this relationship.

H_1: Disclosure of software vulnerability positively influences the actual exploitation

Factors that affect exploitation

The first hypothesis is concerned with the effects of information availability on the actual exploitation of it. Characteristics of the information may have an effect on the actual exploitation. Hansman and Hunt (2005) in their taxonomy of network and computer attacks, present the issues related to vulnerability as the third of the four dimensions they used to classify computer attacks. Fithen, Hernan et al.(2003) in their work on the modeling of vulnerability discuss the precondition and impacts of a security vulnerability. Preconditions may include certain software being used and patches not installed. Impacts are the results of exploitation and it may include privilege escalation, and data alteration.

Based on this, it is argued here that the exploiter who is interested in exploiting the software vulnerabilities weighs in both the preconditions and the impacts. Certain preconditions such as unpatched computer system could be more vulnerable to attacks. Similarly vulnerabilities that cause greater impact could be treated differently by the malicious code writers. Subsequent hypothesis were formed on this premise reflecting the need to consider preconditions and impacts on the actual exploitation of software vulnerability.

Criticality of the Vulnerability:

Not all vulnerabilities are equal with regard to the damage it can do. Potential impact of the vulnerability is given importance in the literature. Jiwnani and Zelkowitz (2002) in their work on issues related to maintaining software applications stress the need for including the impact of the vulnerability on a system as one of the aspect in software maintenance. It gives more incentive to the malicious code writer to exploit a critical vulnerability than vulnerability not so critical. Hence the second hypothesis tests the relationship between the criticality of the vulnerability and the actual exploitation of the vulnerability.

H_2: Criticality of the vulnerability is positively related to the actual exploitation.

Source of Detection:

Identification of vulnerability is done either by outsiders or by the software vendor themselves. There are many security companies and individuals who specialize in the detection of software vulnerabilities. These outside parties follow Organization for Internet Safety's guidelines in communicating the vulnerability information to the software vendor (Applewhite 2004). These guidelines state that outside parties should contact the software vendors before going public about the existing vulnerabilities in software applications. It can be argued that when the vulnerability is detected by an external agency, the chances of more number of people being aware of the vulnerability is high and hence it increases the likelihood of its exploitation.

H_3: Detection of vulnerability by an external agency is positively related to the actual exploitation.

Relatedness of vulnerabilities:

At times, vulnerabilities may be related to one another and vendors try to fix them simultaneously. For example, cumulative vulnerability may affect the same component of the software and the presence of vulnerability may invariably lead to another new vulnerability to be present. Fithen, Hernan et al.(2003) have studied the relationship between multiple vulnerabilities and arrived at a modeling technique to analyze them in totality. Hansman and Hunt (2005) used cumulativeness of vulnerabilities in their taxonomy of computer attacks. When companies find these multiple vulnerabilities, they tend to fix these vulnerabilities with a cumulative patch that fixes all these related vulnerabilities.

H_4: Cumulative (related) vulnerabilities are positively related to the actual exploitation.

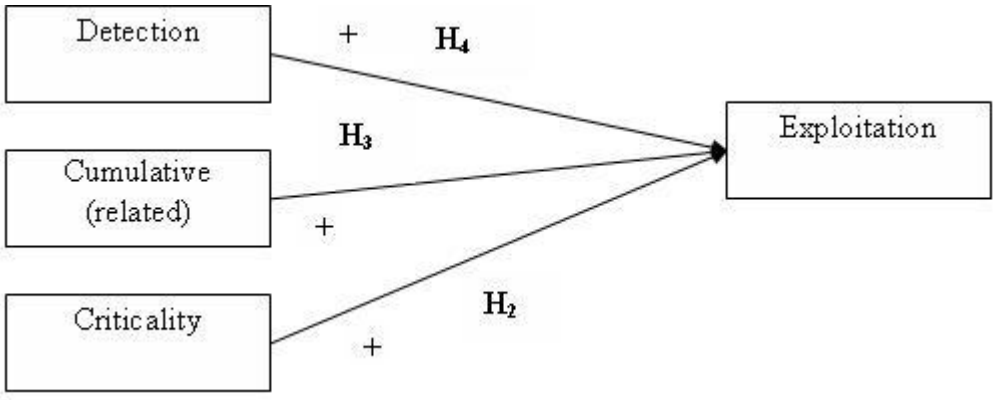

Figure 1. Proposed Research Model

RESEARCH METHODOLOGY

Measures and Data Collection

In order to test the proposed hypothesis, the variables defined were operationalized. Data about computer vulnerabilities are made available by many sources such as Common Vulnerability Exposure (CVE), Bugtraq, CERT/CC advisory, etc. In the past, researches have adopted different approaches, Hansman and Hunt (2005) used vulnerabilities in open-source and proprietary software products and analyzed the differences between them. Fithen, Hernan et al.(2003) used vulnerabilities from Microsoft's Windows in their study, and claimed it to give rich complexity of vulnerabilities that could have interactions with each other. Since there are many software products available from various vendors, vulnerability databases have information spanning many vendors / products. For the purpose of addressing the research question posed in this study, the authors decided to restrict it to a particular software product from a vendor. The software selected for this study is Microsoft's Windows operating system, which commands overwhelming market share of the desktop computers. Choosing a single product gives a homogeneous data and removes influences of extraneous factors and aids in cumulative data analysis.

Data for this study was collected from the Microsoft's security knowledgebase, which contains security bulletins, issued by Microsoft detailing various vulnerabilities. This study's data spanned 3 years i.e. from January 2002 to December 2004. In total hundred and fourteen security bulletins were issued during the study period by the Microsoft Corporation for its Windows operating system. This study primarily considered vulnerabilities in MS Windows and its associated components such as MS Internet Explorer, MS Outlook Express and Windows Media Player. These vulnerabilities are well documented and these bulletins follow standard format in presenting the information. Data from these security bulletins was used for this study. Following table 1 summarizes the number of security bulletins and the actual exploits.

Year	Number of security bulletins	Number of actual exploits
2002	44	13
2003	36	15
2004	34	11
Total	**114**	**39**

Table 1. Microsoft's Security Bulletins and Actual Exploits

Exploitation: Exploitation of software vulnerability is measured using a binary variable that represented either presence of exploitation or not. Exploitation in this case is the presence of virus / worms exploiting a particular vulnerability as described in a security bulletin. Many anti-virus software vendors maintain a comprehensive database documenting the known exploits. Some of the exploits are directly tied to the Microsoft's pre-identified vulnerabilities. Instead of relying on the data from one anti-virus software vendor's data, this study utilized the virus database maintained by multiple anti-virus software vendors.

Following are the virus information databases used in this study: Symantec, McAfee, Trend Micro, and Sophos. These databases provide exhaustive coverage on various known viruses and worms. Microsoft's security bulletin number was checked against these databases. Whenever there was a match, the data corresponding to virus / worm exploits were noted in detail. Some of the security bulletin were not exploited and in other words did not have the corresponding known exploits in the form of virus / worms. Hence the security bulletins were classified either as exploited or not exploited and they were coded accordingly using a binary variable.

Criticality of the vulnerability: Microsoft classifies these vulnerabilities into: critical, important, moderate, and low based on the severity of the damage it can cause. Each security bulletin carries a severity rating. Table 2 presents a classification of the various vulnerabilities that were present in the study data according to their criticality of impact.

Year	Critical	Important	Moderate	Low
2002	26	15	3	1
2003	22	10	2	1
2004	16	15	3	0
Total	**64**	**40**	**8**	**2**

Table 2. Microsoft's Security Bulletins Classified on Criticality

Detection: As mentioned in the previous sections, vulnerabilities in a software product can be detected by end users or by outside security agencies. Microsoft's security bulletin acknowledges the third-party which provided the vulnerability information. A binary variable was used to capture the presence or non-presence of a third party's involvement in the detection of the vulnerability. In total 84 of the 114 vulnerabilities analyzed in this study was detected by outside parties.

Cumulative bulletin: Some of the security bulletins carried information about multiple vulnerabilities. These vulnerabilities could be related or unrelated but they all belong to a particular product. A binary variable is used to represent these cumulative or non-cumulative bulletins. Of the 114 vulnerabilities analyzed in this study 29 of them were of cumulative bulletins.

Year	Detection by outside parties	Cumulative bulletins / patches
2002	29	13
2003	28	6
2004	27	10
Total	**84**	**29**

Table 3. Microsoft's Security Bulletins Classified on Detection / Cumulative

Table 4 illustrates the way in which coding was done. A comprehensive database was created with details about the vulnerabilities and the exploits.

Bulletin No.	Period	Severity	Product / component	Detection / outsiders	Cumulative	Impact	Exploited
MS03-049	Nov. 2003	Critical	Windows	Yes	No	Remote code execution	Yes / W32.Dinfor. Worm
MS04-004	Feb. 2004	Critical	Internet Explorer	Yes	Yes		
MS04-006	Feb. 2004	Important	Windows	Yes	No	Remote code execution	

Table 4. Coded Microsoft's Security Bulletins with Virus Information

Data Analysis

The collected data about the vulnerabilities and the exploits were mostly categorical. The use of parametric techniques such as: regression / analysis of variance are not suitable for testing the formulated hypotheses. Hence non parametric and special parametric techniques that allow the use of categorical variables were used in the study. Next section presents the results along with the statistical tests carried out.

RESULTS AND DISCUSSION

This exploratory study can address the issues related to making vulnerability information available to public and help in better understanding of this phenomenon. Testing of various hypotheses was done using appropriate statistical methods.

H_1: Information about the vulnerability and exploitation

In this particular study we had 114 reported vulnerabilities and 39 actual exploitations. If the exploitation is due to chance then, the probability of occurrence of exploitation vs. non-exploitation of vulnerability is equally split. Since the data was categorical, Chi-square test for equal proportion was used. This test specifically tested if the probability of occurrence / non-occurrence of an exploit is the same.

$\chi 2 = 11.368$

Critical value = 3.841 at $\alpha = 0.05$ for k =1

The computed chi-square value is 11.368 whereas the critical value at alpha of 5% is 3.841 for one degree of freedom. Hence, the first hypothesis is supported and from the chi-square test we can conclude that the proportions were significantly different. In other words, the differences in the exploitation are not due to chance. In this study it is more in favor of non-exploitation of the vulnerability. The available data does not support the contention that publicly available vulnerability information increases the chance of actual exploitation.

Lack of support to the first hypothesis could be due to various reasons. One of the reasons could be attributed to the type of software namely *operating systems* used in this study. Another reason could be the time difference between the availability of vulnerability information and the actual exploitation of the vulnerability. More research in this area may throw better picture on the relationship between vulnerability information disclosure and the exploitation of it.

H_2, H_3, and H_4: Exploitation vs. criticality, source of detection and relatedness of vulnerability

Rest of the hypotheses were tested using logistic regression procedure. Logistic regression is better suited for H_2 to H_4 and this is due to many reasons. First, the binary dependent variable excludes the use of simple linear regression. Logistic regression is capable of handling binary dependent variable. Second, logistic regression procedure is robust against the violation of assumptions in the sample. This study utilized independent variables that were all categorical. Based on the proposed research model for the vulnerability exploitation, all the three independent variables were simultaneously entered in to the logistic regression model.

Results of the logistic regression analysis supported the hypotheses 2 and 3 at $\alpha = 0.01$ and $\alpha = 0.05$ respectively. Hypothesis 4 was not supported at $\alpha=0.05$. Table 5 provides the results and the classification table. As there were 114 vulnerability bulletins of which 39 were exploited and 75 were not exploited. Classification of exploited and non-exploiters by chance would be:

$$((39/114)^2 + (1-(39/114))^2) = 54.98\%.$$

The logistic regression model achieved a classification accuracy of 73.68%, which is much better than by random choice.

Variable	Regression Coefficient	Standard Error	Chi-Square Beta=0	p-value
Intercept	-4.263	1.287	10.96	0.001
Detection	-0.670	0.498	1.81	0.178
Criticality	1.117	0.340	10.79	0.001*
Cumulative	0.976	0.495	3.88	0.049*

Observed	Predicted	
	Exploited	Not Exploited
Exploited	25	16
Not Exploited	14	59
Percent Correctly Classified = **73.68**		

Table 5. Results of logistic regression analysis

Implications for the Practice

First hypothesis that tested the differences between the probabilities in the occurrence of virus exploits were same was not supported and this shows that the information availability about the presence vulnerability did not affect the actual exploitation. Results of this study also provide an indication that the criticality of the vulnerability and cumulativeness of vulnerability are related to the actual exploitation. Though it may be intuitive to assume the importance of the criticality of the vulnerability, this study provided an empirical support to it. The results from this study points to the need for installing patches of high criticality. Installation of the critical patches may thwart virus attacks that exploit the vulnerability. Same is the case for cumulativeness of vulnerabilities.

Limitations

This study is exploratory in nature and the focuses on the products offered by only one vendor (Microsoft). Not all the hypotheses were supported in this study. The proposed model needs to be refined in order to consider other factors such as the type of application, type of attack (Worms / Trojans) etc. Since only publicly available information is used, unreported vulnerabilities as well as their exploits were not studied.

FUTURE RESEARCH DIRECTIONS AND CONCLUSION

Future Research Directions

This study used data for three years and it pertained to one software vendor's product. In future, the scope of this study could be increased to include software from other vendors and this will increase the genralizability of the results. This study excluded the characteristics of the exploit such as the type of virus / worms, time lag for actual exploitation, and the severity of impact and its relation to the vulnerability characteristics. It would be interesting to see the relationship between them. In the future, detailed analysis of vulnerabilities could be done in relation to the type of application exploited such as operating system, database management system, etc.

Conclusion

Practitioner press often highlights the growing economic loss due to the virus attacks. There are fears that zero-day attacks i.e. vulnerability detection and attacks on the same day, are going to be prevalent in the near future. Under this scenario this study made an attempt to relate the availability of vulnerability information and the actual exploitation of the vulnerabilities. This study provides an idea as to how the information availability affects the vulnerability exploitation. Though some of the studied hypotheses were not supported, there exists strong support for the criticality, and relatedness of the vulnerability to be important in the actual exploitation of the vulnerability. Future studies in this area can greatly help in finding the relationship between vulnerability information disclosure and the actual exploitation of it.

REFERENCES

1. Applewhite, A. (2004) In the news - Whose bug is it anyway? The battle over handling software flaws, *IEEE Software*, 21, 2, 94-97
2. Arora, A., Krishnan, R., Nandkumar, A., Telang, R., and Yang, Y. "Impact of Vulnerability Disclosure and Patch Availability - An Empirical Analysis," Economics and Information Security, The Third Annual Workshop on, University of Minnesota, 2004.
3. CSI-FBI "Computer Crime and Security Survey," Computer Security Institute, San Francisco, CA.
4. Fithen, W.L., Hernan, S.V., O'Rourke, P.F., and Shinberg, D.A. (2003) Formal Modeling of Vulnerability., *Bell Labs Technical Journal*, 8, 4, 173-186
5. Furnell, S. (2004) When vulnerability reports can work against us, *Network Security*, 2004, 6, 11-15
6. Goth, G. (2004) How useful are attack trend resources?, *Security & Privacy Magazine, IEEE*, 2, 2, 9-11
7. Hansman, S., and Hunt, R. (2005) A taxonomy of network and computer attacks, *Computers & Security*, 24, 1, 31-43
8. Jiwnani, K., and Zelkowitz, M. "Maintaining software with a security perspective," Proceedings. International Conference on Software Maintenance, 2002, pp. 194-203.
9. Krsul, I.V. (1998) Software vulnerability analysis,
10. Martin, R.A. "Managing vulnerabilities in your commercial-off-the-shelf (COTS) systems using an industry standards effort," Digital Avionics Systems Conference, 2002. Proceedings. The 21st, 2002, pp. 4A1-1-4A1-13 vol.11.
11. Nachenberg, C. (1997) Computer Virus -- Antivirus Coevolution., *Communications of the ACM*, 40, 1, 46-51
12. Panko, R.R. (2003) Slammer: The first blitz worm, *Communications of AIS*, 2003, 11, 207-218
13. Reinke, J., and Saiedian, H. (2003) The availability of source code in relation to timely response to security vulnerabilities, *Computers & Security*, 22, 8, 707-724
14. Venter, H.S., and Eloff, J.H.P. (2004) Vulnerability forecasting--a conceptual model, *Computers & Security*, 23, 6, 489-497
15. Whitman, M.E. (2003) Enemy at the Gate: Threats to Information Security, *Communications of the ACM*, 46, 8, 91